Rules

Kindly strictly follow these instructions in upcoming classes:

❖**Arrive on time***(within first three minutes)* **for the class to minimize disruptions**. No students will be entertained for break at scheduled lecture time.

❖Sit as per Roll No. wise **from Left side of the classroom bench.**

❖**Verify the list of absentees** after completion of class attendance . Student is only responsible for this.

❖. **Silence Devices:** Put mobile phone in silent mode and keep them inside the bag.

❖**Active listening:** Pay close attention to the teacher during lecture.

❖Avoid unnecessary side conversations, when you have any question raise your hand and wait for teacher's acknowledgement.

❖**Issues resolution:** If you have a disagreement or issue, address it through appropriate channels, such as speaking to the subject teacher or school authorities.

❖Dear students, If anyone getting any problem in practice assignment, post your problem in LPU Live CSE320 group.

❖Come up with your charged laptop in CSE320 classes whenever required.

LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

Lecture 0

The Kick Start Session

LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

Vision

To be a globally recognized school through excellence in teaching, learning and research for creating *Computer Science professionals*, *leaders and entrepreneurs of future* contributing to society and industry for sustainable growth.

Mission

- To build computational skills through hands-on and practice-based learning with measurable outcomes.
- To establish a strong connect with industry for in-demand technology driven curriculum.
- To build the infrastructure for meaningful research around societal problems.
- To nurture future leaders through research-infused education and lifelong learning.
- To create smart and ethical professionals and entrepreneurs who are recognized globally

Program Educational Objectives (PEO)

- The graduates shall demonstrate professional advancement through expanded leadership capabilities and technical accomplishment providing solutions to local and global societal issues through mindful engagement.
- The graduates shall undertake higher education or global certifications or exhibit impactful research accomplishment.
- The graduates shall extend global expertise in technology development and deployment by becoming an entrepreneur, consultant and innovator.
- Graduates shall embrace ethics and lifelong learning to adapt to a fast-changing world and enhance global employability in diverse work environments.

LOVELY PROFESSIONAL UNIVERSITY

Program Outcomes (PO)

NAAC GRADE A++

Outcome	Heading	Description
PO1	Engineering knowledge	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem analysis	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions	Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations
PO6	The engineer and society	Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability	Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics	Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work	Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance	Demonstrate knowledge and understanding of the engineering, management principles and apply the same to one's own work, as a member or a leader in a team, manage projects efficiently in respective disciplines and multidisciplinary environments after consideration of economic and financial factors.
PO12	Life-long learning	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSO)

- Apply acquired skills in software engineering, networking, security, databases, intelligent systems, cloud computing and operating systems to adapt and deploy innovative software solutions for diverse applications.
- Apply diverse IT skills to design, develop, and evaluate innovative solutions for business environments, considering risks, and utilizing interdisciplinary knowledge for efficient real-time projects benefiting society.

Course Outcomes

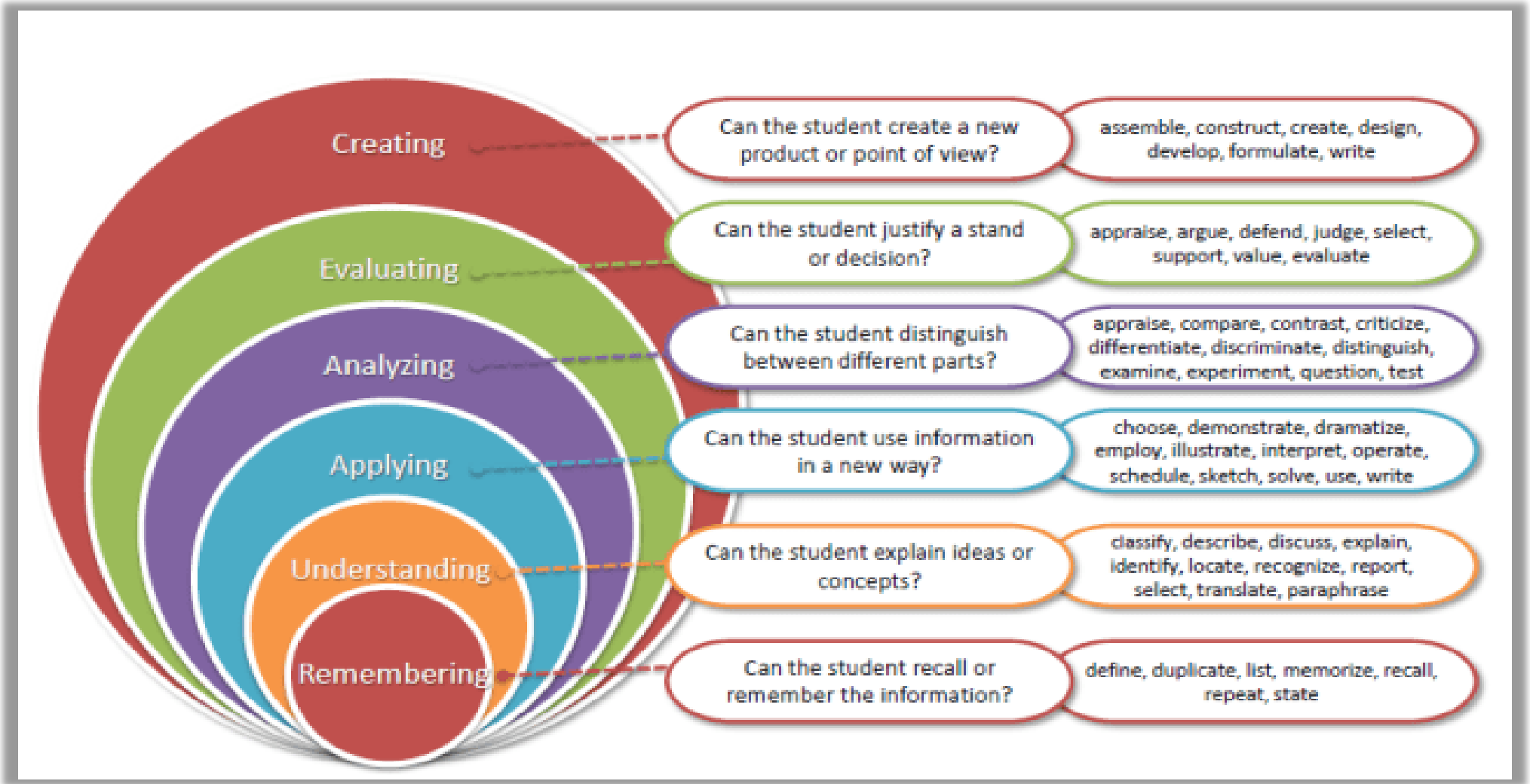
- CO1: understand various software development life cycle models and illustrate software requirement specification
- CO2: construct software design from requirement specification by following structured and organized process.
- CO3: apply the constructs of unified modelling language (UML) for object modelling.

Course Outcomes

- CO4: explain the fundamentals of testing, levels of testing and various types of testing techniques.
- CO5: assess project progress using project management techniques.
- CO6: examine various software quality standards and current trends in the area of software engineering.

Outcomes

Revised Bloom's Taxonomy

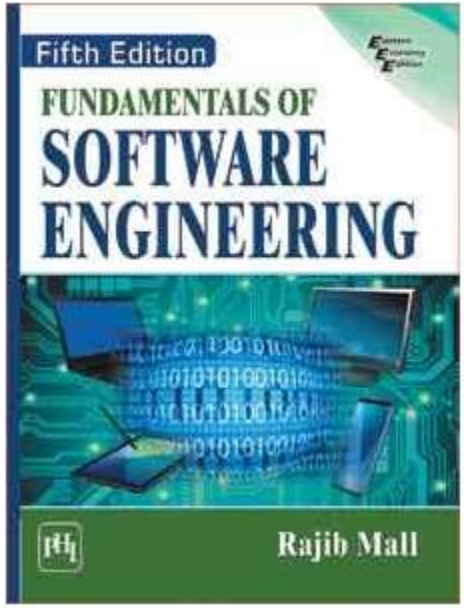


Course details

- **Course Title –** Software Engineering
- **Course Code –** CSE 320
- **LTP –** 3 0 0 [Three lectures/week]
- **Credits –** 3

- **Text Book**

  FUNDAMENTALS OF SOFTWARE ENGINEERING by RAJIB MALL, PHI (PRENTICE HALL INDIA),



LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

Course Assessment Model

Marks break up*

• Attendance	05
• CA(Assignment(Case Based)+Test(MCQ))	25
• MTT (MCQ-Based)	20
• ETE (MCQ-Based)	50
Total	**100**

LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

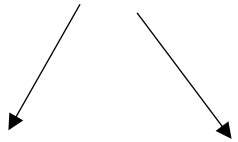
Detail of Academic Tasks

- **AT1: **Assignment- Case based**(Real life problem based)
- **AT2: ***Test(MCQ* **Based Class Test)**

(AT1 and AT2, both are compulsory)

Assessment Model

- CA Category- A0202
- Exam Category- 11



MTT is purely objective

ETE is purely objective

- Evaluation Rubrics

AT1 (Assignment(Case Based): **SRS Document as per IEEE SRS format**-30%, **System design**-30% , **Test cases**-20% and **viva**-20 %

AT2 (Test(MCQ based class test): **MCQ**-30marks

The course contents

Before MTE

- **Introduction to software engineering** :

  Evolution and impact of software engineering, ***Software life cycle models Waterfall model, Prototyping model, Evolution and Spiral models,*** **Feasibility study,** Functional and non-functional requirements, Requirement gathering, Requirement analysis and specification, DevOps, CI/CD pipeline

- **Issues in software design** :

  Basic issues in software design, Modularity, Cohesion, Coupling and Layering, Function oriented s/w design, DFDs and Structure Chart, Microservices Architecture, Layered Architecture, Serverless Computing

- **Object modelling** :

  Object modelling using UML, Object oriented software development, User interface design, Coding standards and code review techniques

LOVELY PROFESSIONAL UNIVERSITY

The course contents

NAAC GRADE A++

After MTE

- **Testing** :

  Fundamentals of testing, black box testing techniques, White box testing techniques, Levels of testing, Test Cases, API testing, Performance testing, Security Testing, AI Testing

  Introduction to Selenium: Feature & Versions of selenium, Record and Playback

- **Software project management** :

  Project management, Project planning and control, Cost estimation, Project scheduling using PERT and GANTT charts, Software Configuration Management, Overview of GitHub Actions, Jenkins, GitHub CI/CD

- **Quality management** :

  Quality management, ISO and SEI CMMI, PSP and Six sigma, Software Maintenance, reuse, CBSD, CASE, Advance topics of Software Engineering.

Online Education Resource (OER) Details

Unit1:

https://www.cs.uct.ac.za/mit_notes/software/pdfs/SE_top.pdf

Unit2:

https://www.cs.uct.ac.za/mit_notes/software/pdfs/SE_top.pdf

Unit3:

https://www.cs.uct.ac.za/mit_notes/software/pdfs/SE_top.pdf

Unit4:

https://www.cs.uct.ac.za/mit_notes/software/pdfs/SE_top.pdf

Unit5:

https://archive.mu.ac.in/myweb_test/MCA%20study%20material/M.C.A%20(Sem%20-%20IV)%20Paper%20-%20Software%20Project%20Management.pdf

Unit6:

http://csis.pace.edu/~marchese/SE616_New/L12/Ch24_summary.pdf
https://www.cprime.com/wp-content/uploads/woocommerce_uploads/2013/05/agile-scrum-glossary.pdf

CSE320: Software Engineering course earns its status as a star course

- because it is pivotal for building a strong foundation in software development, aligning with industry needs, and preparing students for diverse, high-demand roles. Its emphasis on real-world applications and modern trends ensures students graduate as competent, future-ready professionals.

Execution Strategy

Topic Activities	Details of the Activities Planned	Tentative week of conduct of activity	Responsibility	Expected Outcome	References	Quantification
CA1-Case study allocation	Case study assigned on a specific topic	2nd week	Individual Teacher	Students will able to learn about the SRS, system design, test cases	Self-Prepared	1
Quiz	Students will be giving quiz on different software development models	4th week	Individual Teacher	Students will able to learn about the software development models and their selection criteria	Self-Prepared	1

Cont.

Topic Activities	Details of the Activities Planned	Tentative week of conduct of activity	Responsibility	Expected Outcome	References	Quantification
Design Sprint	A function-oriented design sprint would be conducted where students will be assigned scenario-based question to design function-oriented design – **DFD**	5th Week	Individual teacher	Student will able to design DFD for given scenario	Self-Prepared	1
AT2-MCQ based test	To evaluate the student through MCQ Based test	6th week	Individual teacher	Topics covered from week 1 to week 4 for test. Lifecycle models:10 Marks, SRS:10 marks, DFDs and UML:10 marks	Self-Prepared	1
Workshop	An object-oriented design workshop would be conducted where students will be assigned scenario-based question to design function-oriented design – **Use Case diagram**	7th week	Individual teacher	Student will able to design use case diagram for given scenario	Self-Prepared	1

LOVELY PROFESSIONAL UNIVERSITY

Cont.

NAAC GRADE A++

Topic Activities	Details of the Activities Planned	Tentative week of conduct of activity	Responsibility	Expected Outcome	References	Quantification
Live Demonstration On Software	Live demonstration on Automation Testing tool **Selenium** would be presented for the better understanding of software testing	8th week	Individual teacher	Students will able to understand importance of software testing	Self-Prepared	1
Workshop	1 hours Workshop in which students will be given **sample websites** to perform the testing assigned system.	9th week	Individual teacher	Students will apply testing techniques for testing the assigned websites	Self-Prepared	1
Guest Lecture	Industrial expert will discuss and demonstrate **the use of testing technologies based upon currents trends** of industries	10th week	Individual teacher	Students will able to learn about the current technologies and current trends from the industrial expert.	Industrial Expert	2

LOVELY PROFESSIONAL UNIVERSITY

Cont.

NAAC GRADE A++

Topic Activities	Details of the Activities Planned	Tentative week of conduct of activity	Responsibility	Expected Outcome	References	Quantification
Presentation	Students will be giving presentations on different **quality models**	11th week	Individual teacher	Students will able to learn about the quality system and their selection criteria	Self-Prepared	1
Case Study: SRS Activity	A system (like UMS, Flipkart etc.) will be assigned to each student in the beginning of the term and students will write complete SRS, do system design and write test cases for the assigned system.	12nd week Submission	Individual teacher	Student will learn and write SRS/test-cases and develop the DFDs of the assigned system	Self-Prepared	1

Note: Maintain proofs for all the activities

LOVELY PROFESSIONAL UNIVERSITY

Pedagogical Initiatives as Star Course

NAAC GRADE A++

Lecture Number	Pedagogical Initiative	Expected Outcomes	References	Quantification
13	Live Demonstration on Data Flow Diagrams using relevant tool.	Student will learn the working on industry graded tools	Self-Prepared	1
16	Live Demonstration on UML diagrams using relevant tool	Student will learn the working on industry graded tools	Self-Prepared	1
20	Role-play activity related to industry-specific problems	Student will be experience industry related problem	Self-Prepared	1
28	Gamification on current Affairs related to Software Engineering	Student will have discussion on current trends related to testing and other Aspects of Software Engineering	Self-Prepared	1
10,19,30,36	Quizzes on different topics can be conduct on interactive platforms like Kahoot.	Students will have a interactive revision based quiz which will provide them revision on different topics, summary report will also help teacher to create new strategy for students.	Self-Prepared	4
39	Workshop on Selenium Testing tool	A workshop will be conducted for the hand-on experience on the software testing tools	Self-Prepared	1

CA Description

Type of Assessment	Type of Assessment e.g. Test/Term Paper/project etc.	Details of academic Task	Parameters/Rubric of Evaluation	Marks	Allotment/ submission week
CA1	Assignment-Case Based	Students will write complete SRS, do system design and write test cases for the assigned system.	SRS as per format-30%, System design-30% , Test cases-20% and viva-20 %	30	2/12
CA2	MCQ-Based Test	MCQ based test of 30 marks to prepare students for competitive exams.	Lifecycle models & SRS:10 marks, Estimations:10 marks, DFDs & UML:10 marks	30	5/6

LOVELY PROFESSIONAL UNIVERSITY

Gamification

NAAC GRADE A++

CA 1: Gamification approach is used for the CA 1 on basis of milestones mentioned. Week 5: SRS Document as per IEEE SRS format Week 7: System Design Week 10: Test Cases Week 12: Final Submission • The process is designed to make the learning and submission engaging by incorporating game elements like levels, points, and badges. • Students progress through stages such as stakeholder analysis, drafting functional and non-functional requirements, and creating diagrams (e.g., UML). • Badges are awarded for milestones, while peer reviews and leader boards foster competition and collaboration. • Tools like Trello or Class craft can track progress. This approach enhances understanding, motivates students, and ensures timely submission of high-quality SRS documents.	**Rubrics:** • SRS Document as per IEEE SRS format: 30% • System Design: 30% • Test Cases: 20% • Viva: 20%	The following Badges will be given to the students: **Initiator** – Awarded for selecting the project topic and identifying stakeholders. **Analyzer** – For completing stakeholder analysis and understanding user needs. **Innovator** – For proposing creative and unique functional/non-functional requirements. **Visualizer** – For crafting accurate and visually clear diagrams (e.g., UML, DFD). **Strategist** – For completing the feasibility study with well-researched insights. **Timekeeper** – For submitting the SRS on or before the deadline. **Collaborator** – For contributing effectively to team discussions and peer reviews. **Champion** – For overall excellence in completing the SRS with the highest score.

Note:

- Use text-books and reference books for subject preparation
- Use OER(Online education resources)
- Fill time to time feedbacks

The questions in mind…

- **Why the course is being taught?**
- **Where the concepts learned here would be applicable?**

Why the course is being taught?

The course ensures students:

- Understand the software development lifecycle from planning to maintenance.
- Gain hands-on skills in design, modelling, testing and project management.
- Stay current with emerging trends in software engineering to enhance their employability and innovation potential.
- By covering both foundational concepts and modern advancements, **CSE320** bridges academic theory with real-world practices, preparing students for a successful career in software engineering.

Where the concepts learned here would be applicable?

- By mastering the concepts taught in CSE320, students are prepared to contribute effectively to software projects, whether in **established companies**, **startups**, **research environments**, or **independent ventures**.

Real-World Scenarios

- **Startup Projects**:
  - Building a new product or MVP (Minimum Viable Product) using Agile methodologies.
- **Enterprise Applications**:
  - Managing large-scale software systems using PERT and GANTT charts.
- **Open-Source Contributions**:
  - Using GitHub and CI/CD pipelines to collaborate on open-source projects.
- **Freelancing**:
  - Gathering client requirements, designing solutions, and delivering software with industry best practices.

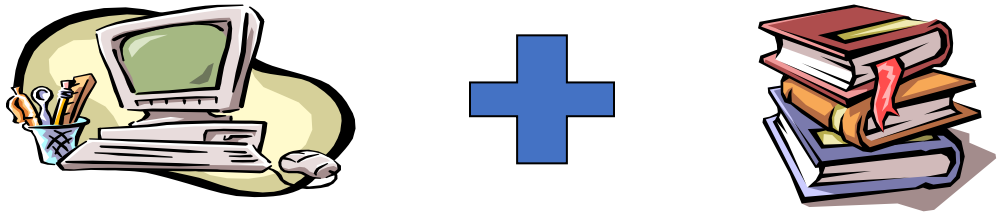
The hitch…

The BURNING questions in mind…

- **What is software? Is it different from Program?**
- **What is Software Engineering?**
- **Why Software Engineering?**
- **What are learning outcomes?**

What is software?

- **Computer programs** and **associated documentation**



- **Software products** may be developed for a particular customer or may be developed for a general market.
- **Software products** may be

1. **Generic** - developed to be sold to a range of different customers

2. **Bespoke** - developed for a single customer according to their specification

Software Components

- **Program**

A program is a subset of software and it becomes software only if documentation and an operating procedure manual are prepared.

- **Documents**

Software documentation consist all the description, programs, graphics and instructions pertaining to design, coding, testing and preparation of software.

- **Operating Procedures(User Manual & Operational Manual)**

Provides information about what software is; how to work with it; how to install it on your system; and how to control all the activities of software

Software crises

- The various software crises are:

1. Over-budget.
2. Not delivering product on time.
3. Product is of poor quality.
4. Software product is not meeting the customer requirements.

Software Engineering

The **software** is a collection of integrated programs along with proper documentation and user manuals

Engineering is the application of **scientific** and **practical** knowledge to **invent, design, build, maintain**, and **improve frameworks, processes, etc**.

What is software engineering?

Software engineering is an engineering discipline which is concerned with all aspects of software production

Software engineers should

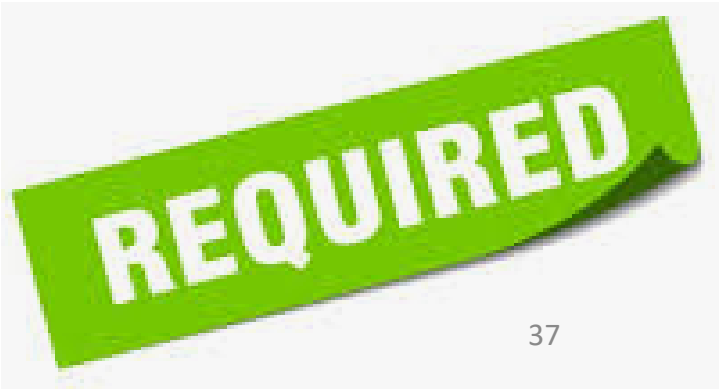
- adopt a systematic and organised approach to their work
- use appropriate tools and techniques depending on
  - the problem to be solved,
  - the development constraints and
  - the resources available

Phases of Development

Why is Software Engineering required?

Software Engineering is required due to the following reasons:

- To manage Large software
- For more Scalability
- Cost Management
- To manage the dynamic nature of software
- For better quality Management



The Fundamental Role of Software Engineering-1

A bridge from **customer needs** to **programming implementation**

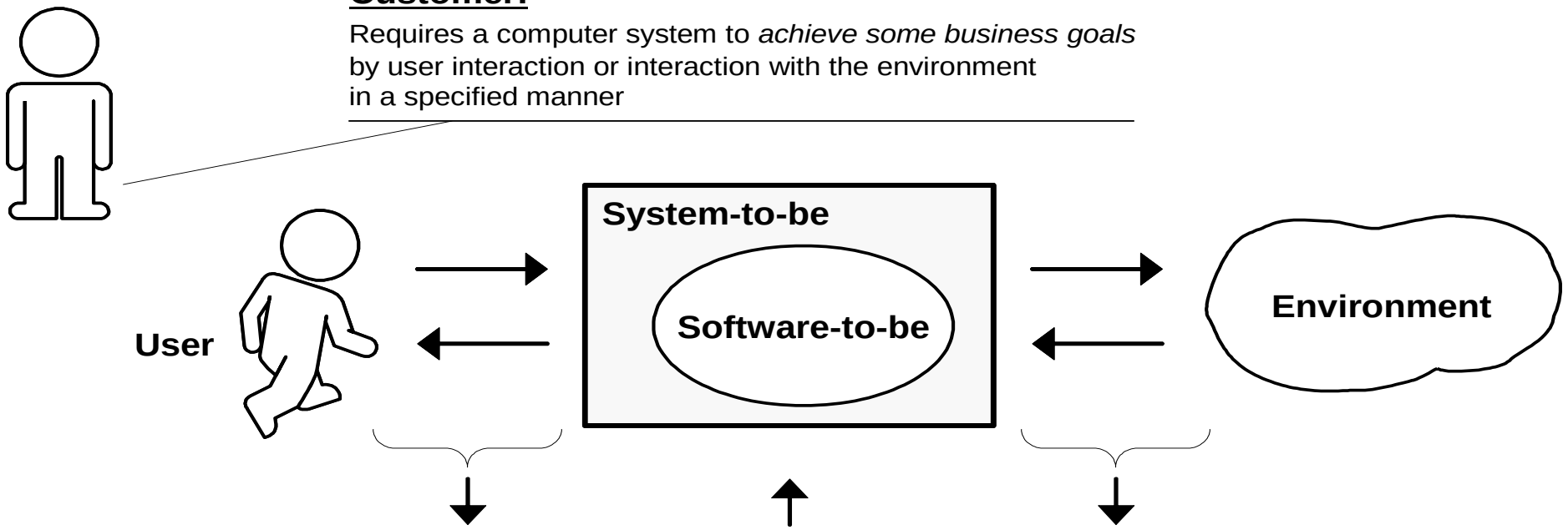
- transforming customer requirements into working software solutions.

First law of software engineering :

"You cannot solve it, unless you understand it."

Software engineer must learn the problem domain (problem cannot be solved without understanding it first)

The Role of Software Engineering-2



Customer:

Requires a computer system to *achieve some business goals* by user interaction or interaction with the environment in a specified manner

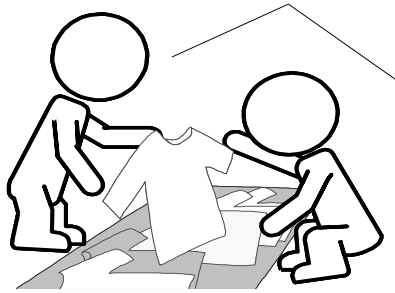
Software Engineer's task:

To ***understand how*** the system-to-be needs to interact with the user or the environment so that customer's requirement is met and ***design*** the software-to-be

Programmer's task:

To ***implement*** the software-to-be designed by the software engineer

May be the same person



Example: ATM Machine

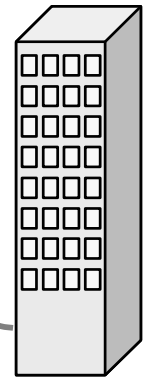
Understanding the money-machine problem:

Bank customer

ATM machine

Communication link

To process customer requests, transfer funds and ensure security

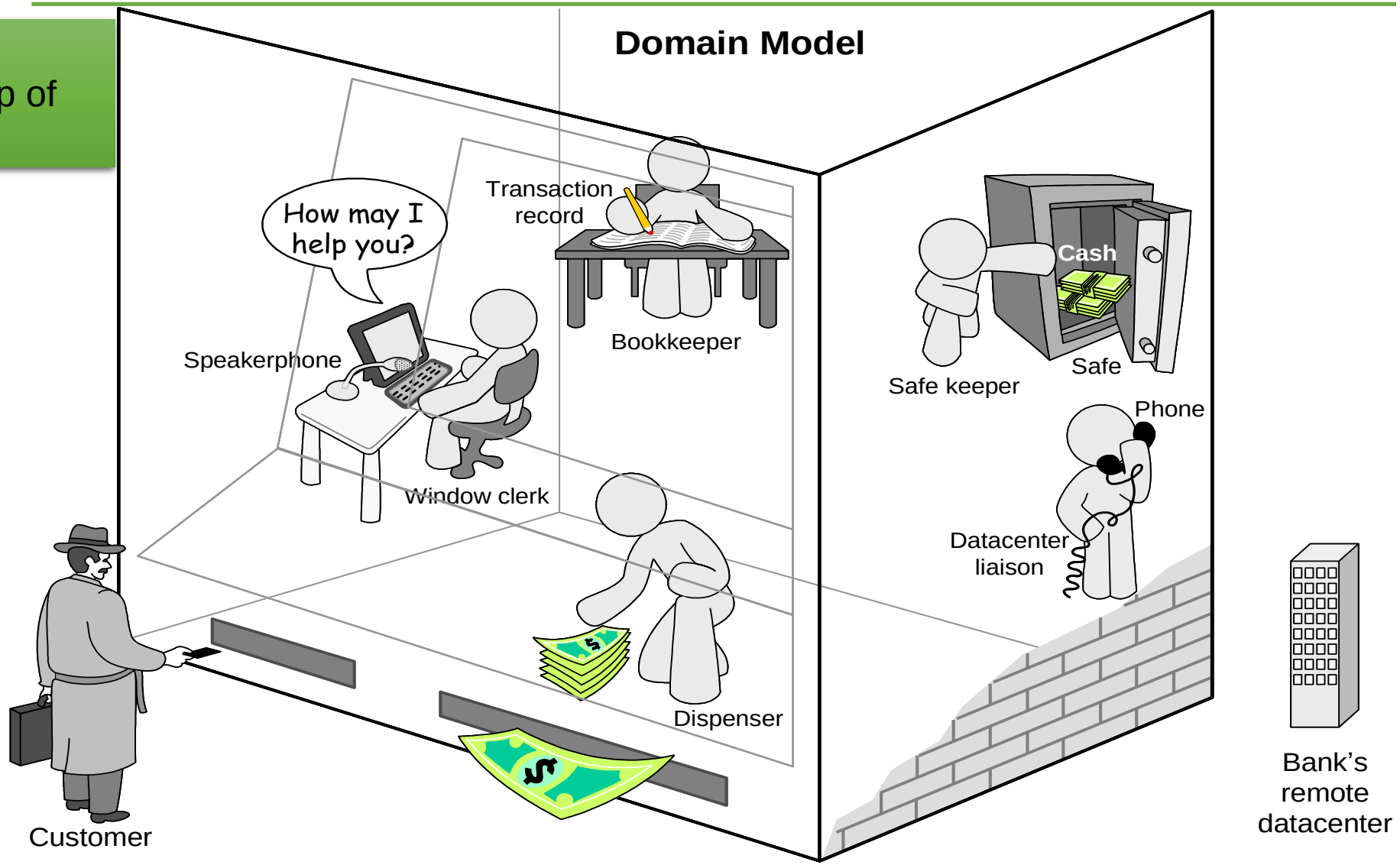


Bank's remote datacenter

Stores customer account details and handles transactions

How ATM Machine Might Work

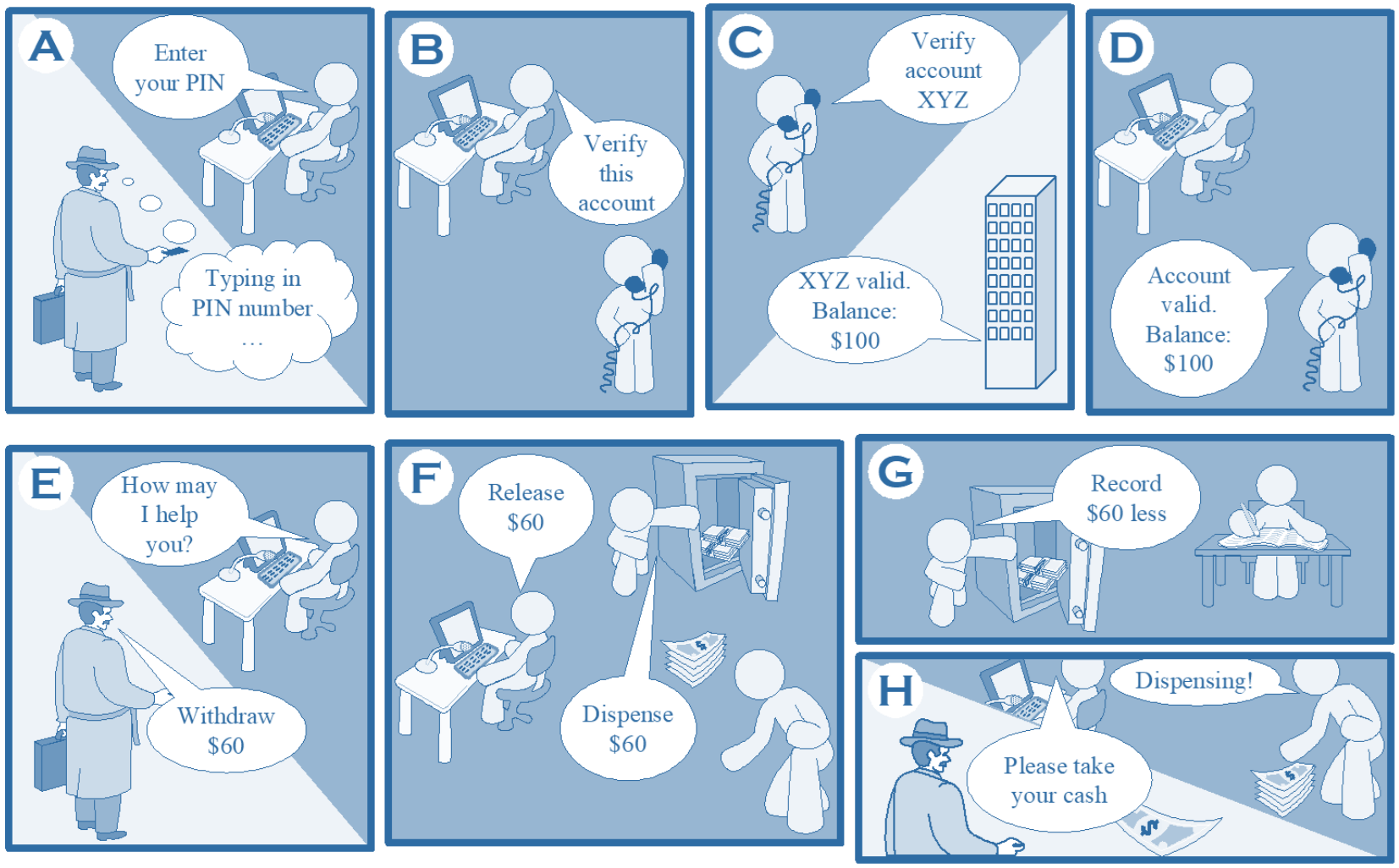
Domain model created with help of domain expert



LOVELY PROFESSIONAL UNIVERSITY

NAAC GRADE A++

Cartoon Strip: How ATM Machine Works?



Software Engineering Blueprints

- *Specifying software problems and solutions is* like cartoon strip writing
- Unfortunately, most of us are not artists, so we will use something less exciting:
  Designing symbols
- However …

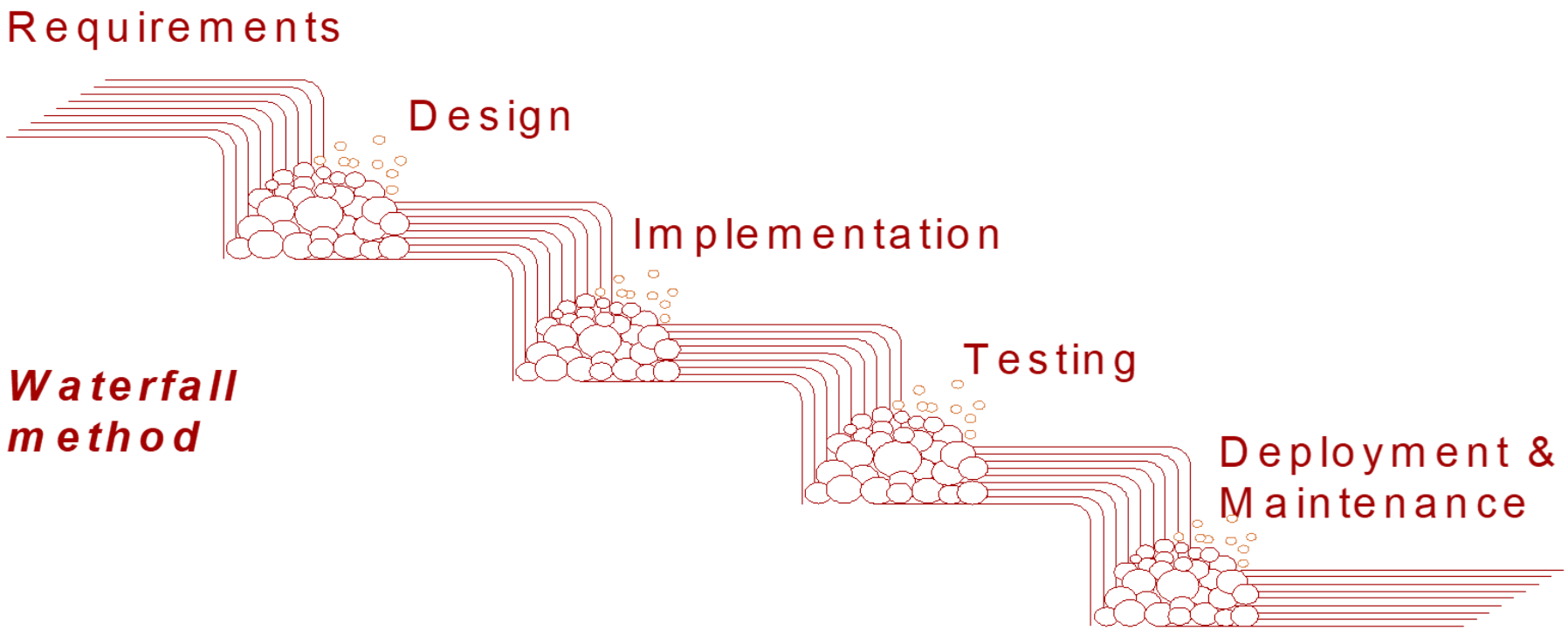
Second Law of Software Engineering

- **Software should be written for people first**
  - (Computers run software, but hardware quickly becomes outdated)
  - Functional (Useful) + good quality(well written) software lives long
  - To nurture software, people must be able to understand it. (This requires clean, clear, and documented code, along with proper design principles.)

Software Development Methods

- Method = work strategy
  - The Feynman Problem-Solving Algorithm:
    (i) Write down the problem (ii) think very hard, and (iii) write down the answer.
- Waterfall
  - Unidirectional, finish this step before moving to the next
- Iterative + Incremental
  - Develop increment of functionality, repeat in a feedback loop
- Agile
  - User feedback essential; feedback loops on several levels of granularity

Waterfall Method



Unidirectional, no way back finish this step before moving to the next

Software myths

1. "If we get behind schedule, we can just add more people"

- Fact: Adding people to a late project makes it even later.
- Someone has to teach the new people.

2. "A general statement of objectives is enough to start programming".

- Fact: Incomplete requirements are a major cause for project failures.

3. "Changes in requirements are easy to deal with because software is flexible".

- Fact: Changes are hard and expensive.
- Especially during coding and after software deployment.

Software myths

4. "Once we get the program running, we are done"

- Fact: Most effort comes after the software is delivered for the first time.
- Bug fixes, feature enhancements, etc

5. "The only product is the running program"

- Fact: Need the entire configuration
- Documentation of system requirements, design, programming, and usage

What are the attributes of good software?

The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable

- **Maintainability**
  - Software must evolve to meet changing needs
- **Dependability**
  - Software must be trustworthy
- **Efficiency**
  - The optimal use of system resources, such as memory and processing power to achieve desired outcomes.
- **Usability**
  - Software must be usable by the users for which it was designed

What are learning outcomes?

By the end of the course, students will be able to:

- Design, develop, test, and manage software projects effectively.
- Align their work with industry standards and trends.
- Tackle software engineering challenges in both academic and professional settings.

Next Class: Evolution and Impact of software engineering, Software Life Cycle Models